

09/662,399MS147164.1REMARKS

Claims 1-45 are currently pending in the present application and are presently under consideration. A clean version of all pending claims is found at pages 2-12. Favorable reconsideration is requested in view of the comments herein.

I. Rejection of Claims 16, 23, 30, 32-35, 37, 38, 41, 42, 43, and 45 under 35 U.S.C. §102(b)

Claims 16, 23, 30, 32-35, 37, 38, 41, 42, 43, and 45 stand rejected under 35 U.S.C. §102(b) as being anticipated by Oppenheim (US 5,734,905). Reconsideration and allowance of these is respectfully requested for at least the following reasons. Oppenheim does not teach or suggest each and every feature of applicants' invention as recited in the subject claims.

For a prior art reference to anticipate, 35 U.S.C. §102 requires that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (quoting *Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)).

Regarding independent claims 16, 32, and 42, Oppenheim does not disclose a *function object* utilized to ...*create a mapping... between a source object and a target object* as recited in these claims. The present invention facilitates constructing a mapping between two disparate object types without requiring an ordinary user to have extensive knowledge in computer programming. The claimed *function object* facilitates *creation of a mapping* between two different object types by graphically associating *a source object node* to a *target object node* via a graphical component of the *function object*.

In contrast, Oppenheim teaches transforming an object into a disparate object utilizing a pre-defined mapping. Oppenheim discloses facilitating communication between two objects, wherein one object can be transformed into another object (or modified by another object) if a mapping between the two objects has been scripted prior to a user requesting such transformation (e.g., two disparate objects communicate with

09/662,399

MS147164.1

one another to determine if a mapping between such two objects presently exists). For example, in an instance that a user desires transformation of a document of type A into a document of type B using the Oppenheim invention, computer code for a mapping between document A and document B must be written and compiled prior to the request of transformation by the user (See Col. 5, lines 38-44, requiring a set of transformation scripts). Oppenheim discloses that an error message occurs when transformation scripts are unavailable between selected objects (See Col. 5, lines 45-50). As a mapping between objects has already been created, no *function object* can be utilized to facilitate creation of a mapping between disparate object types.

Oppenheim further discloses linking applications (application program objects) to various other applications to effectuate operation as a single application (See col. 8, lines 23-26). The application program objects receive a flow of data and manipulate the data as it passes through such application program objects ("The purpose of the object linking process is to cause data flowing through one object to be automatically routed to another object for further processing") See col. 8, lines 34-36. For example, the embodiment illustrated in Fig. 8 links an A/D Converter 270 to a Filter 276 via a Signal Processor 272. A stream of data over time is required for such application objects (e.g., the A/D Converter 270, Signal Processor 272, and Filter 276) to operate effectively. The application program objects modify the continuous stream of data and pass the data onto a disparate application program object. The application program objects disclosed in Oppenheim, however, differ from objects that are mapped to one another as recited in these claims. The present invention facilitates creation of a mapping between a *source object and a target object*, wherein a *node* of the source object can be associated with a *node* of the target object via the *function object*. *Nodes* are components defining objects (e.g., an object can be defined by a plurality of nodes). Therefore, a particular mapping between two disparate objects can be created by a computer user not an expert in computer programming. In contrast to the source and target objects of the present invention, Oppenheim discloses linking applications that do not contain nodes to facilitate flow of data between the linked applications. Thus a particular mapping between the applications (and nodes of the applications if such nodes exist) must be generated prior to linking the applications.

09/662,399

MS147164.1

As described *infra*, the invention disclosed in Oppenheim facilitates transfer of data between two or more applications (not objects as recited in the subject claims), wherein the first application has a data output port and the second application has a data input port. If a mapping between two applications has been *previously scripted*, data output from the first application can be delivered to the input port of the second application. The process can then be repeated to direct data through several applications, wherein mapping between each pair of applications has been *previously scripted*. In the example of Fig. 8, Oppenheim teaches associating three applications, wherein a first mapping *previously scripted* facilitates transfer of data from the first and second applications, and a second mapping *previously scripted* facilitates transfer of data between the second and third applications. The second application does not become a *function object* utilized to generate a particular mapping between the first and third application as recited in the subject claims. Rather, the second application operates without regard to the first and third applications (e.g., the second application manipulates data delivered from the first application as it would to any input data). Therefore, Oppenheim does not disclose a *function object* utilized to associate a target object with a source object as recited in these claims.

Regarding independent claims 33 and 41, as Oppenheim does not disclose a *function object* employed to create a mapping between a *source object* and a *target object*, Oppenheim cannot disclose a system and method for creating a *function object*. Further regarding claims 33 and 41, as well as independent claims 16, 32, 33, 42, 43 and 45, Oppenheim does not teach *creating a script component for performing a function*. Oppenheim discloses utilizing “transformation scripts” for transforming one object into another object (see col. 5 lines 38-41 and Fig. 2). Such “transformation scripts” do not perform a function as claimed in the present invention - they are an entire mapping between two object types. Applicant respectfully disagrees with the Examiner’s contention that Fig. 5 and corresponding description related thereto disclose a *script component for performing a function* in connection with *creating a mapping between a source object and a target object*. Oppenheim discloses that a first object (transformer object) can be “slapped” onto a second object (transformee object) to generate a new or revised object. However, there is not a source object and a target object as recited in

09/662,399MS147164.1

these claims. – an entirely new object (or a revised version of the same object) is generated (e.g., there is no target object that is mapped to the transformee object). Furthermore, the program application objects illustrated in Fig. 8 likewise do not comprise *script component for performing a function* in connection with *creating a mapping between a source object and a target object*. The program application objects simply manipulate a flow of data and pass such data along to a disparate application, and thus do not facilitate creation of a mapping between two objects. In contrast, the script component as recited in these claims facilitates creating a user-specified mapping between two disparate objects.

In view of at least the above, it is respectfully submitted that Oppenheim does not anticipate independent claims 16, 32, 33, 41, 42, 43, and 45, and claims 23, 30, 34, 35, 37, and 38 which depend therefrom.

II. Rejection of Claims 1-15, 17-22, 24-27, and 44 under 35 U.S.C. §103(a)

Claims 1-15, 17-22, 24-27, and 44 are rejected under 35 U.S.C. §103(a) as being unpatentable over Oppenheim and Microsoft's "Component Object Model Specification" (COM Specification). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Neither Oppenheim nor Microsoft's COM specification alone or in combination teach or suggest all the claim limitations of the subject invention.

To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. See MPEP §706.02(j).

Regarding independent claims 1, 16 and 44, as discussed *supra*, Oppenheim does not teach or suggest a *function object* utilized to facilitate creation of a mapping between two objects, let alone particular components of the *function object*.

09/662,399

MS147164.1

Microsoft's COM specification fails to make up for the deficiencies of Oppenheim *vis a vis* applicants' claimed invention, and therefore the subject rejection should be withdrawn.

III. Rejection of Claims 28, 29, and 31 under 35 U.S.C. §103(a)

Claims 28, 29, and 31 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Oppenheim in view of Jordan (US 5,778,227). Reconsideration and allowance of these claims is respectfully requested for at least the following reasons. Jordan does not make up for the aforementioned deficiencies of Oppenheim regarding independent claim 16.

As claim 16 is believed to be in condition for allowance, the rejection of the subject dependent claims is moot. It is respectfully requested that this rejection be withdrawn.

IV. Rejection of Claims 36, 39, and 40 under 35 U.S.C. §103(a)

Claims 36, 39, and 40 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Oppenheim, and also over Faustini (US 6,496,870). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons.

Faustini does not make up for the aforementioned deficiencies of Oppenheim regarding independent claim 33, rendering the subject rejection moot. Withdrawal of the rejection is therefore respectfully requested.

09/662.399MS147164.1**V. Conclusion**

The present application is believed to be condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063.

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,

AMIN & TUROCY, LLP



Himanshu S. Amin
Reg. No. 40,894

AMIN & TUROCY, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731